Pastebin-Lite Application

Gulshan Gupta
guptagulshan884@gmail.com

January 2, 2026

Abstract

Pastebin-Lite is a Java Spring Boot application that allows users to create, share, and view text pastes. Pastes can expire automatically based on time (TTL) or the number of views. The application uses Java, Spring Boot, Spring Web, Spring Data JPA, Thymeleaf, and MySQL. It follows a clean Controller-Service-Repository architecture with REST APIs and a simple frontend UI.

1 Features

- Create and view text pastes
- Optional TTL (Time-to-Live in seconds) per paste
- Optional max view count per paste
- Automatic expiration with 404 handling
- REST API with proper HTTP responses
- Deterministic testing using custom headers
- Simple HTML/Thymeleaf UI

2 Technology Stack

- **Backend:** Java 17, Spring Boot, Spring Web, Spring Data JPA
- **Database:** MySQL 8+
- **Frontend:** HTML, Thymeleaf, JavaScript
- **Build Tool:** Maven

3 Project Structure

```
com.kodnest.pastebin

 controller       # REST controllers (PasteController, HealthController)
 service          # Business logic (PasteService)
 repository       # Data access (PasteRepository)
 entity           # Database entities (Paste.java)
 dto              # Data transfer objects (optional)
 exception        # Custom exceptions and global handler
 util             # Utilities (TimeUtil)
 PastebinLiteApplication.java
```

4 Database Design

Column	Type	Description
id	VARCHAR	Primary key (short unique ID)
content	TEXT	Text of the paste
created_at	BIGINT	Epoch milliseconds of creation
ttl_seconds	BIGINT	Optional TTL in seconds
max_views	INT	Optional max view count
current_views	INT	Number of times the paste was viewed

5 REST API Endpoints

5.1 Create Paste

POST /api/pastes

Request Body:

```
{
  "content": "Hello World",
  "ttlSeconds": 3600,
  "maxViews": 5
}
```

Response (201 Created):

```
{
  "id": "abc12345",
  "url": "/api/pastes/abc12345"
}
```

5.2 Fetch Paste

GET /api/pastes/id

Optional Header:

```
X-CURRENT-TIME: 1700000000000
```

Response (200 OK):

```
{
  "content": "Hello World"
}
```

Errors:

- 404 if paste expired or not found

6 UI Pages

- **Create Paste:** create.html
- **View Paste:** paste.html

The UI uses JavaScript to call REST APIs to create and view pastes.

7 Setup & Run

1. Install MySQL and create database:

   ```
   CREATE DATABASE pastebin;
   ```

2. Configure `application.properties`:

   ```
   spring.datasource.url=jdbc:mysql://localhost:3306/pastebin
   spring.datasource.username=root
   spring.datasource.password=yourpassword
   spring.jpa.hibernate.ddl-auto=update
   spring.jpa.show-sql=true
   ```

3. Run the application:

   ```
   mvn spring-boot:run
   ```

4. Open browser at:

   ```
   http://localhost:8080/create.html
   ```

8 Testing

- Unit tests for service layer (TTL, max views, expired pastes)
- Integration tests for controller endpoints
- Deterministic time testing using header `X-CURRENT-TIME`

9 How It Works

1. User creates a paste via UI or API
2. Paste is stored in MySQL with optional TTL and max views
3. When fetching a paste:
    - TTL and max views are checked
    - If expired or exceeded, 404 is returned
    - Otherwise current_views is incremented and content returned

10 Key Takeaways

- Clean Controller-Service-Repository architecture
- REST APIs with proper HTTP status codes
- Expiration logic for TTL & max views
- Supports deterministic testing
- Simple and functional frontend UI

Author

Gulshan Gupta
Email: `guptagulshan884@gmail.com`